



**Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»
Курс «Базовые компоненты интернет-технологий»
Отчет по домашнему заданию
Тестирование Telegram бота

Выполнил:
Студент группы ИУ5-34Б
Данилин Максим
Дата и подпись:

Преподаватель:
Гапанюк Ю.Е.
Дата и подпись:

2021 г.

Постановка задачи

1. Модифицируйте код лабораторной работы №6 таким образом, чтобы он был пригоден для модульного тестирования.
2. Используя материалы лабораторной работы №4 создайте модульные тесты с применением TDD - фреймворка (2 теста) и BDD - фреймворка (2 теста).

Текст программы Файл config.py

```
from enum import Enum

TOKEN = '5096949774:AAENOQUddYBbibu1-WauVAx_cyMdk6Qqa-A'

db_file = 'db.vdb'

CURRENT_STATE = "CURRENT_STATE"

class States(Enum):
    START_STATE = "START_STATE"
    GENRE_STATE = "TYPE_STATE"
    CLASSIC_STATE = "CLASSIC_STATE"
    ROCK_STATE = "ROCK_STATE"
    XIP_XOP_STATE = "XIP_XOP_STATE"
    SONG_STATE = "SONG_STATE"
```

Файл dbworker.py

```
from vedis import Vedis
import config

def get(key):
    with Vedis(config.db_file) as db:
        try:
            return db[key].decode()
        except KeyError:
            return config.States.START_STATE.value

def set(key, value):
    with Vedis(config.db_file) as db:
        try:
            db[key] = value
            return True
        except:
            return False

def make_key(message_id, state):
    return "{}_{}".format(message_id, state)
```

Файл functions.py

```
import os

cur_path = os.getcwd()
```

```
def path(genre, author, song):
    return os.path.join(cur_path, '{}\{}\{}.mp3'.format(genre, author, song))
```

Файл bot.py

```
import telebot
from telebot import types
import config
import dbworker
from functions import path

bot = telebot.TeleBot(config.TOKEN)

@bot.message_handler(commands=['start'])
def welcome(message):
    markup = types.ReplyKeyboardMarkup(row_width=1, resize_keyboard=True)
    markup.add(types.KeyboardButton('Привет, хорошо'))
    dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE),
config.States.GENRE_STATE.value)
    bot.send_message(message.chat.id, 'Привет, я помогу помочь выбрать песню',
reply_markup=markup)

@bot.message_handler(func=lambda message:
dbworker.get(dbworker.make_key(message.chat.id, config.CURRENT_STATE)) ==
config.States.GENRE_STATE.value)
def genre(message):
    if message.text in ['Классический', 'Хип-хоп', 'Рок']:
        if message.text == 'Классический':
            dbworker.set(dbworker.make_key(message.chat.id, config.States.GENRE_STATE.value),
'Классический')
            dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE),
config.States.CLASSIC_STATE.value)
            button1_1 = types.KeyboardButton('Иоганн Себастьян Бах')
            button2_2 = types.KeyboardButton('Людвиг ван Бетховен')
            button3_3 = types.KeyboardButton('Петр Ильич Чайковский')
        elif message.text == 'Хип-хоп':
            dbworker.set(dbworker.make_key(message.chat.id, config.States.GENRE_STATE.value),
'Хип-хоп')
            dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE),
config.States.XIP_XOP_STATE.value)
            button1_1 = types.KeyboardButton('50 cent')
            button2_2 = types.KeyboardButton('Drake')
            button3_3 = types.KeyboardButton('Kanye West')
        elif message.text == 'Рок':
            dbworker.set(dbworker.make_key(message.chat.id, config.States.GENRE_STATE.value),
'Рок')
            dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE),
config.States.ROCK_STATE.value)
            button1_1 = types.KeyboardButton('AC DC')
            button2_2 = types.KeyboardButton('Nirvana')
            button3_3 = types.KeyboardButton('Queen')
    markup1 = types.ReplyKeyboardMarkup(row_width=3, resize_keyboard=True)
    markup1.add(button1_1, button2_2, button3_3)
```

```

        bot.send_message(message.chat.id, 'Выберите автора: ',
reply_markup=markup1)
    else:
        markup = types.ReplyKeyboardMarkup(row_width=3, resize_keyboard=True)
        button1 = types.KeyboardButton('Классический')
        button2 = types.KeyboardButton('Хип-хоп')
        button3 = types.KeyboardButton('Рок')
        markup.add(button1, button2, button3)
        bot.send_message(message.chat.id, 'Выберите жанр:', reply_markup=markup)

@bot.message_handler(func=lambda message:
dbworker.get(dbworker.make_key(message.chat.id, config.CURRENT_STATE)) ==
config.States.CLASSIC_STATE.value)
def classic_author(message):
    if message.text in ['Иоганн Себастьян Бах', 'Людвиг ван Бетховен', 'Петр Ильич
Чайковский']:
        if message.text == 'Иоганн Себастьян Бах':
            dbworker.set(dbworker.make_key(message.chat.id, 'author'), 'Иоганн Себастьян
Бах')
            dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE),
config.States.SONG_STATE.value)
            button1_1 = types.KeyboardButton('Менуэт сюита №2')
            button2_2 = types.KeyboardButton('Сарабанда')
            button3_3 = types.KeyboardButton('Шутка')
        elif message.text == 'Людвиг ван Бетховен':
            dbworker.set(dbworker.make_key(message.chat.id, 'author'), 'Людвиг ван Бетховен')
            dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE),
config.States.SONG_STATE.value)
            button1_1 = types.KeyboardButton('Adagio Sostenuto')
            button2_2 = types.KeyboardButton('Allegrettto')
            button3_3 = types.KeyboardButton('Allegro Con Brio')
        elif message.text == 'Петр Ильич Чайковский':
            dbworker.set(dbworker.make_key(message.chat.id, 'author'), 'Петр Ильич
Чайковский')
            dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE),
config.States.SONG_STATE.value)
            button1_1 = types.KeyboardButton('Вальс Цветов')
            button2_2 = types.KeyboardButton('Танец маленьких лебедей')
            button3_3 = types.KeyboardButton('Щелкунчик')
        markup1 = types.ReplyKeyboardMarkup(row_width=3, resize_keyboard=True)
        markup1.add(button1_1, button2_2, button3_3)
        bot.send_message(message.chat.id, 'Выберите песню: ',
reply_markup=markup1)
    else:
        markup = types.ReplyKeyboardMarkup(row_width=3, resize_keyboard=True)
        button1 = types.KeyboardButton('Иоганн Себастьян Бах')
        button2 = types.KeyboardButton('Людвиг ван Бетховен')
        button3 = types.KeyboardButton('Петр Ильич Чайковский')
        markup.add(button1, button2, button3)
        bot.send_message(message.chat.id, 'Выберите автора:', reply_markup=markup)

```

```

@bot.message_handler(func=lambda message:
dbworker.get(dbworker.make_key(message.chat.id, config.CURRENT_STATE)) ==
config.States.ROCK_STATE.value)
def rock_author(message):
    if message.text in ['AC DC', 'Nirvana', 'Queen']:
        if message.text == 'AC DC':
            dbworker.set(dbworker.make_key(message.chat.id, 'author'), 'AC DC')
            dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE),
config.States.SONG_STATE.value)
            button1_1 = types.KeyboardButton('Back in Black')
            button2_2 = types.KeyboardButton('Highway to Hell')
            button3_3 = types.KeyboardButton('Thunderstruck')
        elif message.text == 'Nirvana':
            dbworker.set(dbworker.make_key(message.chat.id, 'author'), 'Nirvana')
            dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE),
config.States.SONG_STATE.value)
            button1_1 = types.KeyboardButton('In Bloom')
            button2_2 = types.KeyboardButton('Lithium')
            button3_3 = types.KeyboardButton('Smells Like Teen Spirit')
        elif message.text == 'Queen':
            dbworker.set(dbworker.make_key(message.chat.id, 'author'), 'Queen')
            dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE),
config.States.SONG_STATE.value)
            button1_1 = types.KeyboardButton('Bohemian Rhapsody')
            button2_2 = types.KeyboardButton('I Want to Break Free')
            button3_3 = types.KeyboardButton('We Will Rock You')
        markup1 = types.ReplyKeyboardMarkup(row_width=3, resize_keyboard=True)
        markup1.add(button1_1, button2_2, button3_3)
        bot.send_message(message.chat.id, 'Выберите песню: ',
reply_markup=markup1)
    else:
        markup = types.ReplyKeyboardMarkup(row_width=3, resize_keyboard=True)
        button1 = types.KeyboardButton('AC DC')
        button2 = types.KeyboardButton('Nirvana')
        button3 = types.KeyboardButton('Queen')
        markup.add(button1, button2, button3)
        bot.send_message(message.chat.id, 'Выберите автора:', reply_markup=markup)

@bot.message_handler(func=lambda message:
dbworker.get(dbworker.make_key(message.chat.id, config.CURRENT_STATE)) ==
config.States.XIP_XOP_STATE.value)
def xip_xop_author(message):
    if message.text in ['50 cent', 'Drake', 'Kanye West']:
        if message.text == '50 cent':
            dbworker.set(dbworker.make_key(message.chat.id, 'author'), '50 cent')
            dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE),
config.States.SONG_STATE.value)
            button1_1 = types.KeyboardButton('Candy Shop')
            button2_2 = types.KeyboardButton('In Da Club')
            button3_3 = types.KeyboardButton('PIMP')
        elif message.text == 'Drake':
            dbworker.set(dbworker.make_key(message.chat.id, 'author'), 'Drake')

```

```

        dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE),
config.States.SONG_STATE.value)
        button1_1 = types.KeyboardButton('Life Is Good')
        button2_2 = types.KeyboardButton('Nonstop')
        button3_3 = types.KeyboardButton('Toosie Slide')
    elif message.text == 'Kanye West':
        dbworker.set(dbworker.make_key(message.chat.id, 'author'), 'Kanye West')
        dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE),
config.States.SONG_STATE.value)
        button1_1 = types.KeyboardButton('I Am A God')
        button2_2 = types.KeyboardButton('New Slaves')
        button3_3 = types.KeyboardButton('Stronger')
    markup1 = types.ReplyKeyboardMarkup(row_width=3, resize_keyboard=True)
    markup1.add(button1_1, button2_2, button3_3)
    bot.send_message(message.chat.id, 'Выберите песню: ',
reply_markup=markup1)
else:
    markup = types.ReplyKeyboardMarkup(row_width=3, resize_keyboard=True)
    button1 = types.KeyboardButton('50 cent')
    button2 = types.KeyboardButton('Drake')
    button3 = types.KeyboardButton('Kanye West')
    markup.add(button1, button2, button3)
    bot.send_message(message.chat.id, 'Выберите автора:', reply_markup=markup)

@bot.message_handler(func=lambda message:
dbworker.get(dbworker.make_key(message.chat.id, config.CURRENT_STATE)) ==
config.States.SONG_STATE.value)
def song(message):
    with open(path(dbworker.get(dbworker.make_key(message.chat.id,
config.States.GENRE_STATE.value))),
            dbworker.get(dbworker.make_key(message.chat.id, 'author')), message.text),
'rb') as music:
        bot.send_audio(message.chat.id, music)
    dbworker.set(dbworker.make_key(message.chat.id, config.CURRENT_STATE),
config.States.GENRE_STATE.value)
    markup = types.ReplyKeyboardMarkup(row_width=1, resize_keyboard=True)
    markup.add(types.KeyboardButton('Попробовать еще раз'))
    bot.send_message(message.chat.id, 'Продолжим?', reply_markup=markup)

if __name__ == '__main__':
    bot.infinity_polling()

```

Файл TDDtest.py

```

import unittest
from functions import path

class Test(unittest.TestCase):
    def test1(self):
        self.assertEqual(path('Рок', 'AC DC', 'Back in Black'), 'D:\бот\Рок\AC DC\Back in
Black.mp3')
    def test2(self):
        self.assertEqual(path('Хип_хоп', 'Kanye West', 'Stronger'),
'D:\бот\Хип_хоп\Kanye West\Stronger.mp3')

```

```

def test3(self):
    self.assertEqual(path('Классический', 'Иоганн Себастьян Бах', 'Менуэт сюита №2'),
                    'D:\бот\Классический\Иоганн Себастьян Бах\Менуэт сюита №2.mp3')

if __name__ == '__main__':
    unittest.main()

```

Файл BDDsteps.py

```

from behave import given, when, then
from functions import path

@given("I have the data {genre}, {author} and {song}")
def process_data(context, genre, author, song):
    context.genre = genre
    context.author = author
    context.song = song

@when("I find the song")
def find_path(context):
    context.path = path(context.genre, context.author, context.song)

@then("I expect the result to be {path}")
def result(context, path):
    assert context.path == path

```

Файл BDDtest.feature

Feature: tests

Scenario: test1

```

Given I have the data Классический, Петр Ильич Чайковский and Щелкунчик
When I find the song
Then I expect the result to be D:\бот\Классический\Петр Ильич
Чайковский\Щелкунчик.mp3

```

Scenario: test2

```

Given I have the data Рок, Queen and We Will Rock You
When I find the song
Then I expect the result to be D:\бот\Рок\Queen\We Will Rock You.mp3

```

Scenario: test3

```

Given I have the data Хип_хоп, Drake and Nonstop
When I find the song
Then I expect the result to be D:\бот\Хип_хоп\Drake\Nonstop.mp3

```

Результат выполнения

Командная строка

```
D:\бот>python TDDtest.py
...
-----
Ran 3 tests in 0.001s

OK

D:\бот>behave BDDtest.feature
Feature: tests # BDDtest.feature:1

  Scenario: test1                                # BDDtest.feature:3
    Given I have the data Классический, Петр Ильич Чайковский and Щелкунчик      # steps/BDDsteps.py:4
    When I find the song                         # steps/BDDsteps.py:10
    Then I expect the result to be D:\бот\Классический\Петр Ильич Чайковский\Щелкунчик.mp3 # steps/BDDsteps.py:14

  Scenario: test2                                # BDDtest.feature:8
    Given I have the data Рок, Queen and We Will Rock You          # steps/BDDsteps.py:4
    When I find the song                           # steps/BDDsteps.py:10
    Then I expect the result to be D:\бот\Rок\Queen\We Will Rock You.mp3 # steps/BDDsteps.py:14

  Scenario: test3                                # BDDtest.feature:13
    Given I have the data Хип_хоп, Drake and Nonstop      # steps/BDDsteps.py:4
    When I find the song                         # steps/BDDsteps.py:10
    Then I expect the result to be D:\бот\Хип_хоп\Drake\Nonstop.mp3 # steps/BDDsteps.py:14

1 feature passed, 0 failed, 0 skipped
3 scenarios passed, 0 failed, 0 skipped
9 steps passed, 0 failed, 0 skipped, 0 undefined
Took 0m0.003s
```